Application No: 10/706,516
Office Action mailed: August 27, 2008
Reply dated: November 26, 2008

In the Claims:

Please amend claims 1 and 10; and add new claims 28-36, as shown below.

1.   (Currently Amended):   A computer program product including a storage medium with instructions thereon for execution by a computer for high level dynamic code generation, the instructions comprising:

a) computer code for dynamically generating at run-time automatically creating a class file container object that stores source code describing a class and selecting a class name and a super class for the class;

b) computer code for adding a first source code defining a method to the class stored in the class file container object, wherein an application programming interface is used to define the method, creating a class file container object includes selecting a class name and a super class for the class;

c) computer code for repeating step b for each method of the class;

d) computer code for adding a second source code into the method in the class stored in the class file container object, wherein an application programming interface is used to define code added into the method;

[[d]] e) computer code for repeating step d instructions b and c to populate each method of the class stored in the class file container object;

[[e]] f) computer code for generating a tree of statements and expressions based on the class stored in the class file container object, wherein each statement and expression is represented as an object, wherein each statement maintains state of the program being generated;

[[f]] g) computer code for using the tree of statements and expressions to generate byte code for the class; and

[[g]] h) computer code for instantiating an instance of the new class file object from the byte code; wherein the computer program product can dynamically generate code.

2 – 9.   Canceled

10.   (Currently Amended):   The computer program product of claim 1 wherein the computer code implements an adapter adaptor class.

11.   (Previously Presented):   The computer program product of claim 1 wherein the computer code implements a proxy class.

12.   (Previously Presented): The computer program product of claim 1 further comprising computer code for:

- 2 -

Attorney Docket No.: ORACL-01316US2
tplunkett/oracl/1316us2/1316us2_Reply_082708.doc
PAGE 3/10 * RCVD AT 11/26/2008 8:20:14 PM [Eastern Standard Time] * SVR:USPTO-EFXRF-6/11 * DNIS:2738300 * CSID:7033590094 * DURATION (mm-ss):02-08

Application No: 10/706,516
Office Action mailed: August 27, 2008
Reply dated: November 26, 2008

repeatedly adding a method to the class stored in the class file container object for each method associated with a stub generated for a remote object.

13. (Previously Presented): The computer program product of claim 12 wherein the computer code for repeatedly adding a method to the class stored in the class file container object for each method associated with a stub generated for a remote object includes program code for:
determining a number of methods associated with the stub in a remote interface.

14 - 15. Canceled

16. (Previously Presented): The computer program product of claim 1 wherein the tree of statements and expressions represents at least one method, the at least one method comprising at least one of: a code statement, an expression, a variable and a programming construct.

17. (Previously Presented): The computer program product of claim 1 wherein the tree of statements and expressions forms a known structure or interface when the class is a known type.

18. (Previously Presented): The computer program product of claim 17 wherein the tree of statements and expressions forms a known structure when the class is at least one of an adapter and a proxy type.

19 - 24. Canceled

25. (Previously Presented): The computer program product of claim 1, wherein the dynamically generated code is used for remote method invocation skeletons, remote method invocation stubs, wrappers for database connections, and proxies used to enforce call-by-value semantics.

26. (Previously Presented): The computer program product of claim 1, wherein dynamically generated code exists for the life of a server it resides upon.

27. (Previously Presented): The computer program product of claim 1, further comprising computer code for generating executable code from the byte code by using a class loader.

28. (New): The computer program product of claim 1, wherein organization of objects in a particular

- 3 -

Application No: 10/706,516
Office Action mailed: August 27, 2008
Reply dated: November 26, 2008

structure or interface avoids a need for a compiler.

29.   (New): The computer program product of claim 1, wherein a statement uses the maintained state to generate an intermediate representation of Java objects that represent individual bytecode instructions.

30.   (New): The computer program product of claim 1, wherein a bytecode assembler converts an intermediate representation into bytecode that can be interpreted by a Java virtual machine.

31.   (New): The computer program product of claim 1, wherein code is added to a method using constructs that correspond to Java language statements, expressions, and variables.

32.   (New): The computer program product of claim 1, wherein after the method is defined in step b, the method is initially empty and contains no code until code is added into the method in step d.

33.   (New): The computer program product of claim 1, wherein an application programming interface is used to define a method or code in the method that is added to the class file container object.

34.   (New): The computer program product of claim 1, wherein dynamic code generation occurs as part of an application server process.

35.   (New): The computer program product of claim 1, wherein the maintained state includes contents of a stack and contents of local variables that are in use at each point of program flow.

36.   (New): A system, comprising:
        one or more processors; and
        an application server including a dynamic code generation module with instructions for execution by the one or more processors, the instructions comprising:
            a) computer code for dynamically generating at run-time a class file container object that stores source code describing a class and selecting a class name and a super class for the class;
            b) computer code for adding a first source code defining a method to the class stored in the class file container object, wherein an application programming interface is used to define the method;
            c) computer code for repeating step b for each method of the class;
            d) computer code for adding a second source code into the method in the class stored in the

- 4 -

class file container object, wherein an application programming interface is used to define code added into the method;

e) computer code for repeating step d to populate each method of the class stored in the class file container object;

f) computer code for generating a tree of statements and expressions based on the class stored in the class file container object, wherein each statement and expression is represented as an object, wherein each statement maintains state of the program being generated;

g) computer code for using the tree of statements and expressions to generate byte code for the class; and

h) computer code for instantiating an instance of the new class file object from the byte code.

- 5 -